

# 基于 MATLAB 的量子粒子群优化算法及其应用\*

余 健<sup>1)</sup> 郭 平<sup>2)</sup>

(韩山师范学院数信学院<sup>1)</sup> 潮州 521041)(北京师范大学信息科学学院<sup>2)</sup> 北京 100875)

**摘 要** 量子粒子群优化(QPSO)算法是在经典的粒子群优化(PSO)算法的基础上所提出的一种具有量子行为的粒子群优化算法,具有高效的全局搜索能力。通过求解 J. D. Schaffer 提出的多峰函数优化问题的实验分析表明,方法具有良好的收敛性和稳定性。

**关键词** QPSO 量子 粒子群

**中图分类号** TP301.6

## 1 引言

经典的粒子群优化(PSO)算法<sup>[3]</sup>是一种基于群体智能的随机搜索算法,具有全局逼近能力,但由于其搜索空间有限,易陷入局部极值。孙俊等人在文献[4]中给出了具有量子行为的粒子群优化算法,即 QPSO 算法。该算法简单有效,收敛速度快,全局搜索性能远优于 PSO 算法。

## 2 粒子群优化算法

PSO 算法首先初始化一群随机粒子,然后通过进化(迭代)找到最优解。每一次进化(迭代)中,粒子通过跟踪两个“极值”来更新自己。一个是粒子本身所能找到的最优解,即个体极值 Pbest,另一个是整个群体目前找到的最优解,即全局极值 Gbest。粒子找到上述两个极值后,就根据下面两个公式更新自己的速度和位置<sup>[6]</sup>:

$$V = W * V + c1 * \text{unifrnd}(0, 1) * (Pbest - X) + c2 * \text{unifrnd}(0, 1) * (Gbest - X) \quad (1)$$

$$X = X + V \quad (2)$$

其中,  $V$  是粒子的速度,  $X$  是粒子的当前位置,  $\text{unifrnd}(0, 1)$  是 0 至 1 之间的随机数,  $c1$  和  $c2$  被称作学习因子,通常被设为 2,  $W$  为加权系数,一般取值在 0.1 至 0.9。粒子通过不断进化,不断搜索,当达到收敛或到达最大进化(迭代)代数的条件后,得到的 Gbest 就是全局最优解。

从动力学的角度看,粒子群算法中的粒子的收敛过程是以 Pbest 点为吸引子,随着速度的减小不

断接近 Pbest 点,最后收敛于 Pbest 点。因此,在整个过程中,在 Pbest 点处实际上存在某种形式的吸引势能场吸引着粒子,这正是整个粒子能保持聚集性的原因<sup>[7]</sup>。但在经典 PSO 算法中,粒子是在经典力学的状态下沿着确定的轨迹飞行,因此粒子搜索的空间是一个有限的区域,因而不能保证一定找到全局最优解。

## 3 量子粒子群优化(QPSO)算法

### 3.1 QPSO 算法的优点

孙俊等人从量子力学的角度,提出一种新的 PSO 算法——量子粒子群优化(QPSO)算法,认为粒子具有量子行为,每一个粒子在搜索空间移动时,存在着一个以 Pbest 为中心的 DELTA 势阱<sup>[4]</sup>。由于在量子空间中的粒子满足聚集态的性质完全不同,粒子移动时没有确定的轨迹,这使粒子可以在整个可行解空间中进行探索寻找全局最优解,因而 QPSO 算法的全局搜索能力远远优于经典的 PSO 算法。

在 QPSO 算法中,设种群规模为  $M$ ,在进化过程中,粒子以一定概率取加或减,更新每个粒子的位置,并生成新的粒子群体,由公式(3)至(6)决定:

$$p = a * Pbest(i) + (1 - a) * Gbest; \quad (3)$$

$$mbest = 1/M * \sum_{i=1}^M Pbest(i); \quad (4)$$

$$b = 1.0 - \text{generation}/\text{maxgeneration} * 0.5; \quad (5)$$

$$\text{if } u > 0.5 \quad (6)$$

$$\text{position} = p - b * |mbest - \text{position}| * \ln(1/u);$$

\* 收到本文时间:2007 年 2 月 9 日

**作者简介:**余健,男,硕士研究生,讲师,研究方向:神经网络、智能计算;郭平,男,教授、博士生导师,IEEE 高级会员,研究方向:模式识别、神经网络等。

else

position = p + b \* |mbest - position| \* ln(1/u);

其中,  $a, u$  为 0 至 1 之间的随机数,  $mbest$  是粒子群  $pbest$  的中间位置, 即平均值,  $b$  为收缩扩张系数, 在 QPSO 收敛过程中线性减小,  $generation$  为当前进化代数,  $maxgeneration$  为设定的最大进化代数。

### 3.2 基于 MATLAB 的仿真

#### 3.2.1 参数编码

在 MATLAB 中, 粒子的位置  $X$  用实数表示。粒子的参数编码格式:  $X_1, X_2, X_3, \dots, X_D$ 。粒子群的编码格式如图 1 所示, 其中  $D$  表示粒子的参数维数, 这由变量的个数决定,  $M$  表示粒子群的规模, 即粒子的个数  $X_{11}, X_{12}, \dots, X_{1D}$  数。

$X_{21}, X_{22}, \dots, X_{2D}$

.....

$X_{M1}, X_{M2}, \dots, X_{MD}$

#### 3.2.2 初始化粒子群

在 MATLAB 中, 对粒子群进行初始化是要生成一个矩阵元素满足图 1 要求的随机矩阵, 另外, 还需要对粒子群个体极值  $Pbest$  和全局极值  $Gbest$  进行初始化。下面给出粒子群初始化伪代码:

POP = unifrnd(xmin, xmax, M, D);

Pbest = POP;

Gbest = unifrnd(xmin, xmax, 1, D);

其中, POP 表示粒子群,  $xmin$  和  $xmax$  分别是变量的下限和上限, unifrnd 是返回  $xmin$  至  $xmax$  之间的随机数。

#### 3.2.3 更新粒子的位置

粒子的位置的更新是基于公式(3)至(6), 下面给出其实现的伪代码:

$a = \text{unifrnd}(0, 1); u = \text{unifrnd}(0, 1);$

$b = 1.0 - \text{当前代数} / \text{最大代数} * 0.5;$

$p = a * Pbest(i, :) + (1 - a) * Gbest;$

$mbest = \text{sum}(Pbest) / M;$

if  $u > 0.5$  POP( $i, :$ ) =  $p - b * \text{abs}(mbest - \text{POP}(i, :)) * \ln(1/u);$

else POP( $i, :$ ) =  $p + b * \text{abs}(mbest - \text{POP}(i, :)) * \ln(1/u);$

end

其中,  $b$  为收缩扩张因子。

#### 3.2.4 主程序

QPSO 算法具体实现步骤如下:

(1) 确定种群规模  $M$  和粒子维数  $D$ , 初始化粒子群体、 $Pbest$  和  $Gbest$ ;

(2) 根据目标函数计算每一个粒子的适应度;

(3) 根据其适应度, 更新个体最优位置  $Pbest$

(i) 和群体最优位置  $Gbest$ ;

(4) 根据公式(3)至(6)以一定概率取加或减, 更新每个粒子的位置, 生成新的粒子群体;

(5) 判断粒子适应度是否满足收敛条件或者是否到达最大进化代数, 是则退出, 否则返回 2)。

## 4 应用实例

基于上述的方法在 MATLAB 中开发了 QPSO 算法工具箱, 并用于求解 J. D. Schaffer 提出的多峰函数优化问题<sup>[6]</sup>:

$\min f(x_1, x_2) =$

$$0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2} \quad (7)$$

s. t.  $-100 \leq x_i \leq 100$

它的全局极小点是 (0, 0), 而在距全局极小点大约 3.14 范围内凹陷部有无数多的局部极小点, 一般算法很难搜索到全局最小点。

用 QPSO 算法, 在配置为 Celeron4 2.0GHZ, 内存 512M 的 PC 机, 进化代数为 100, 运行时间为 7.3079 秒, 就逼近全局极小值, 取得较好结果。

## 5 结束语

实验结果表明, QPSO 算法具有具有良好的收敛性和稳定性。本文开发了基于 MATLAB 的 QPSO 算法工具箱, 并将其应用到典型的函数优化问题上, 取得较好的效果。只要给出目标函数(即可求出粒子的适应度、粒子的参数维数  $D$ ), 设定粒子群的规模  $M$ , 就可以直接利用该算法工具箱求解, 因此具有较强的通用性。

### 参考文献

- [1] 高隽. 神经网络原理及仿真实例[M]. 机械工业出版社, 2003
- [2] 周开利等. 神经网络模型及其 MATLAB 仿直程序设计[M]. 清华大学出版社, 2005
- [3] J. Kennedy, R. Eberhart. Particle swarm optimization [J]. Proceeding IEEE international Conference on Neural Networks, 1995, 4: 1942 ~ 1948
- [4] Jun Sun, Bin Feng, Wenbo Xu. Particle Swarm Optimization with particles having quantum behavior [C]. Congress on Evolutionary Computation, 2004
- [5] 侯志荣等. 基于 MATLAB 的粒子群优化算法及其应用[J]. 计算机仿真, 2003, 20(10): 68 ~ 70
- [6] Schaffer J D, Caruana R A, Eshelman L J, Das R. A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. In [337], 573 ~ 580, 1993